# Developing an automatic part-of-speech tagger for Scottish Gaelic

# Developing an Automatic Part-of-Speech Tagger for Scottish Gaelic

**Samuel Danso**
Celtic and Scottish Studies
University of Edinburgh EH8 9LD
sdanso@staffmail.ed.ac.uk

**William Lamb**
Celtic and Scottish Studies
University of Edinburgh EH8 9LD
w.lamb@ed.ac.uk

## Abstract

This paper describes an on-going project that seeks to develop the first automatic PoS tagger for Scottish Gaelic. Adapting the PAROLE tagset for Irish, we manually re-tagged a pre-existing 86k token corpus of Scottish Gaelic. A double-verified subset of 13.5k tokens was used to instantiate eight statistical taggers and verify their accuracy, via a randomly assigned hold-out sample. An accuracy level of 76.6% was achieved using a Brill bigram tagger. We provide an overview of the project's methodology, interim results and future directions.

## 1 Introduction

Part-of-speech (PoS) tagging is considered by some to be a solved problem (cf. Manning, 2011: 172). Although this could be argued for languages and domains with decades of NLP work behind them, developing accurate PoS taggers for highly inflectional or agglutinative languages is no trivial task (Oravecz and Dienes, 2002: 710). Challenges are posed by the profusion of word-forms in these languages – leading to data sparseness – and their typically complex tagsets (*ibid.*). The complicated morphology of the Celtic languages, of which Scottish Gaelic (ScG) is a member,[1] led one linguist to state, "There is hardly a language [family] in the world for which the traditional concept of 'word' is so doubtful" (Ternes, 1982: 72; cf. Dorian, 1973: 414). As inauspicious as this may seem for our aims, tagger accuracy levels of 95-97% have been achieved for other morphologically complex languages such as Polish (Acedański, 2010: 3), Irish (Uí Dhonnchadha and Van Genabith, 2006) and Hungarian (Oravecz and Dienes, 2002: 710). In this paper, we describe our effort to build – to the best of our knowledge – the first accurate, automatic tagger of ScG.

Irish is the closest linguistic relative to Gaelic in which substantial NLP work has been done, and Uí Dhonnchadha and Van Genabith's work (2006; cf. Uí Dhonnchadha, 2009) provides a valuable reference point. For them, a rule-based method was the preferred option, as a tagged corpus of Irish was unavailable (Uí Dhonnchadha, 2009: 42).[2] They used finite-state transducers for the tokenisation and morphological analyses, and context-sensitive Constraint Grammar rules to carry out PoS disambiguation (2006: 2241). In our case, after consultation, we decided to adopt a statistical approach. We were motivated by the availability of a pre-existing, hand-tagged corpus of Scottish Gaelic (see Lamb, 2008: 52-70), and our expectation that developing an accurate, rule-based tagger would take us beyond our one-year timeframe.

## 2 Methodology

### 2.1 Annotation

Using an adapted form of the PAROLE Irish tagset (Uí Dhonnchadha, 2009: 224), we manually re-tagged the corpus of ScG mentioned above. Significant conversion was required, as the corpus had been designed for a study of register variation (Lamb, 2008). Currently, 13.5k tokens have been final-



[1] The Goidelic branch includes Scottish Gaelic, Irish and Manx Gaelic. Welsh, Breton and Cornish are part of the Brythonic branch.

[2] Uí Dhonnchadha (2009: 213; cf *ibid*: 42) states her future intention to induce a Brill tagger on a Gold-standard corpus of Irish.



*Proceedings of the First Celtic Language Technology Workshop*, pages 1–5,
Dublin, Ireland, August 23 2014.

ised and used to train and evaluate various tagger algorithms, as described below. Our motivations for adapting the Irish tagset were to facilitate comparisons between Irish and ScG corpora, and to follow emergent *de facto* standards, as recommended in Leech (2005). Although this expedited progress, some tokens could not be easily classified.

Like Irish (cf. Uí Dhonnchadha, 2009: 81), ScG morphology is generally regarded as complex, particularly in the nominal system. Various process can re-shape word-forms, resulting in data sparseness; sparsity is a common issue in NLP work with morphologically-rich languages (Orvecz and Dienes, 2002: 711). These processes include initial consonant mutation (e.g. *c* → *ch*); internal vowel change (e.g. *a* → *oi*); palatalisation of final consonants (e.g. *-at* → *-ait*) and affixation. For example, the singular noun *cearc* ['hen'] declines for case and definiteness as *cearc, chearc, circ, chirc, circe* and *chirce*. The adjective *mall* ['slow'] can be found variably as *mall, mhall, malla, mhalla, moill, mhoill, moille* and *mhoille.*[3] To compound issues, as the language attrites, historically robust distinctions are being levelled or inconsistently observed. Another obstacle was ambiguous function words, such as *a* and *a'*; these can be tagged in various ways,[4] depending on context. There were also a small number of fused forms having multiple grammatical categories: e.g. *cuimhneam* ['I know'],[5] ← *cuimhne* ['knowledge'] + *agam* ['at me']. It was not possible, in all cases, to split these at the tokenisation stage and introducing further complexity to an already involved tagset seemed ill-advised. Therefore, we determined to use concatenation tags (cf. Chungku et al., 2010: 105), e.g. *cuimhneam* ['knowledge at me'] <Ncsfn+Pr1s>. This tag is glossed as: Noun common singular feminine nominative + Pronoun prepositional 1st-person singular.

### 2.2 Tokenisation

A full account of the automatic tokeniser is beyond the scope of this paper. What follows is a brief description of our guiding principles and the manual tokenisation of the training corpus. As a rule, we strove for a 1:1 correspondence between words/punctuation and tokens (1). However, some exceptions were necessary. As illustrated in (2) by the phrase *mu dheireadh* ['at last'], multi-word expressions were tokenised together when they performed an indivisible grammatical function[6] and could not be intersected by another word. Here, we took a slightly different approach from Uí Dhonnchadh (2009: 71-72); our preference was for a low number of MWEs in order to avoid the need for a complicated lexicon.[7] In a few cases, we split words into two or more tokens if a failure to have done so would have negatively impacted the pipeline further on (e.g. during lexicon extraction). In (3), this is illustrated by the word *dh'fhuirich* ['stayed'], which has been split into two tokens, separating the morphophonemic particle *dh'* from the verbal form. As described in Uí Dhonnchadha (2009: 70-71), this obviates duplication in the lexicon (cf. $m'ad_1$ ['my hat'] → $m'_1\ ad_2$).

1) 1 WORD → 1 TOKEN
$\text{“}_1\text{Ò}_2\ cha_3\ robh_4\ e_5\ seo_{6,7}\text{”}_8\ ars'_9\ ise_{10}$ → $\text{"}_1\ \text{Ò}_2\ cha_3\ robh_4\ e_5\ seo_{6\ ,7}\ \text{"}_8\ ars'_9\ ise_{10}$

2) ≥ 2 WORDS → 1 TOKEN
$Bhàsaich_1\ am_2\ fear_3\ mu_4\ dheireadh_5$ → $Bhàsaich_1\ am_2\ fear_3\ mu\ dheireadh_4$

3) 1 WORD →≥2 TOKENS
$Dh'fhuirich_1\ e_2\ ann_3$ → $Dh'_1 fhuirich_2\ e_3\ ann_4$

[3] See Lamb (2008: 197-280) for further details on Gaelic grammar. Many of the same issues are encountered in Irish (see Uí Dhonnchadha, 2009).
[4] The word *a*, for instance, can be variably tagged as a 3rd person masc possessive, a relative PN, a verbal agreement marker, the vocative particle, an interrogative pronoun, a simple preposition and a numerical counting particle.
[5] NB: *cuimhneam* is a fused form consisting of a noun and a prepositional pronoun. Like Russian, Gaelic expresses possession in a locative fashion (e.g. *tha e agam* ['I have it', lit. 'it is at me']; there is no verb of possession.
[6] As defined by the tagset.
[7] However, toponyms were tokenised as MWEs, e.g. *Dùn Èideann* 'Edinburgh' (cf. Uí Dhonnchadha, 2009: 72).

More generally, the corpus was manually divided into clauses, with each clause on a separate line. This was done to provide additional context for automatic tag disambiguation, with clause boundaries used in lieu of 'sentence boundaries' for instantiating the taggers. Clauses are linguistically well-defined structures, whilst sentences are not (Miller and Weiner, 1998: 71).

### 2.3 Tagger Instantiation

The PoS tagging task can be formulated as follows: given a word $w_i$, derived from a sequence of words ($w_i \ldots w_n$), assign the best tag $t_i$, derived from a set of tags, $T=\{t_i..t_n\}$. After our 13.5k token sample had been manually tagged and twice verified, we used it to instantiate two stochastic taggers – bigram HMM (see Huang et al., 2009: 214) and trigram TnT (Brants, 2000: 224) – and a hybrid tagger (Brill, 1992: 112), which combines a stochastic and rule-based method. We employed the principle of *ensemble learning* (Dietterich, 2000: 1), whereby simple statistical PoS tagging algorithms can be usefully employed to improve the precision of more sophisticated algorithms. For comparative purposes, we also included simple unigram, bigram and trigram taggers. Simple n-gram algorithms tend to assign tags based on the most frequent tag sequence of the n-gram as observed in the training set.

On the surface, the HMM and TnT algorithms employ similar approaches to tagging, as both analyse the sequential history of word–tag pairings in a given 'sentence' using Markov Model principles (Ghahramani, 2001: 9). However, the approaches employed by HMM and TnT are somewhat different. HMM is based on first-order Markov Model principles, whereas TnT tends to be based upon second-order ones. Additionally, TnT tends to employ additional features during training, such as capitalisation and suffixes (Brants, 2000: 224). The Brill tagger, on the other hand, is an example of Transformational-Based Learning (Brill, 1992: 112). Like a stochastic tagger, it begins by pairing words with their most likely tags, as observed in the training corpus. This can be done using unigrams, bigrams or trigrams. It then notes where tags are applied incorrectly and attempts to induce corrective rules via various context-sensitive templates (*ibid.*: 113). Finally, it re-tags the corpus according to learnt patterns. A typical template is 'replace $t_1$ with $t_2$ in the context of C'. Some glossed examples from the Gaelic corpus follow:

> 1) **Ug → Q-r** if the tag of words i+1...i+2 is '**V-s**' [token = a]
> *Change the tag for the **agreement marker** to one for a **relative pronoun** if one of the next two words is tagged as a **past-tense verb***
>
> 2) **Tdsm → Tdsf** if the tag of words i+1...i+2 is '**Ncsfn'** [token = a']
> *Change the tag for the **singular, masculine definite article** to one for **the singular, feminine definite article** if one of the following two words is a **singular, feminine noun in the nominative***
>
> 3) **Sa → Tdsf** if the tag of the following word is **'Ncsfn'** [token = a']
> *Change the tag for the **aspectual particle** to one for the **singular, feminine definite article** if the following word is tagged as a **singular, feminine noun in the nominative***

One of the advantages of the Brill tagger over other stochastic approaches is its transparency. With a knowledge of the tagset and target language, its output is easily understood. As seen in the above examples, it is capable of handling the problematic homographs discussed in §2.1.

Eight models, in total, were developed and assessed using the same training and testing set (see Table 1). Since the Brill tagger requires the output of a stochastic tagger before applying inductive methods, as described above, we employed the unigram algorithm as a base. Our ensemble strategy used a *backoff* mechanism, implemented as part of the Natural Language Tool Kit (NLTK) libraries (Bird, 2006: 70). Backoff creates a chain of PoS tagging algorithms that are executed in sequential order, ensuring that if an initial tagger is unable to classify a given token, then that token is passed on to the next tagging algorithm. Two ensemble-based models were developed: Brill (with bigram) and Brill (with trigram). Thus, in addition to using the simple unigram model as an initial stochastic tagger with Brill, we also employed bigram and trigram models. Brill (bigram) passes any untagged token to the unigram tagger, whereas the Brill (trigram), employs the bigram algorithm for untagged tokens and then passes any untagged tokens onto the unigram algorithm. In all cases, these stochastic

stages are followed by the inductive of rules characterising the Brill algorithm. We used the default parameters of all algorithms, apart from one in the Brill algorithm, which defines the number of rules to be learned automatically from the training corpus. This was set to 150, as it optimised performance with the training set (NB: it did not apply to the test set).

We employed the hold-out method to evaluate our models (cf. Acedański 2010). To achieve this, we randomly divided the corpus sample into a 10% 'hold-out' set for evaluation (165 sentences, ~986 tokens), and a 90% 'training' set for model development (1492 sentences, ~12,560 tokens). We assessed the performance of the models by calculating the percentage of correctly assigned PoS tags for each against the manually assigned tags.

## 3 Results

The table below shows the preliminary results.

**Table 1: Preliminary performance comparison of 8 statistical taggers**

| ***Model*** | *Unigram* | *Bigram* | *Trigram* | *HMM* | *TnT* | $Brill_{UNI}$ | $Brill_{BI}$ | $Brill_{TRI}$ |
|---|---|---|---|---|---|---|---|---|
| **Accuracy** | 66.1 | 52.1 | 23.6 | 74.6 | 76.1 | 75.6 | **76.6** | 75.2 |

As seen in Table 1, the most successful method, at present, is the Brill bigram model, which had a performance level of 76.6%. This is to be expected given the granularity of the tagset, along with the restricted training data; we expect accuracy to increase once we utilise the full corpus of ~86k tokens.[8] Unsurprisingly, due to sparsity issues, the least successful model was the simple trigram, at 23.6%. The performance of the TnT model was somewhat better than HMM (HMM: 74.6% and TnT: 76.1%), and also better than the Brill unigram model (TnT: 76.1% and $Brill_{UNI}$: 75.6%). The Brill bigram model, which is ensemble-based, outperformed the TnT model by about 0.5% ($Brill_{BI}$: 76.6% and TnT: 76.1%). There was, however, a drop in performance of about 1.4% between the Brill bigram (76.6 %) and Brill trigram (75.2%). Overall, our top accuracy level is comparable to that reported in Dandapat et al. (2007: 223) for their 10k sample (84.73%), although they experienced less sparsity as their tagset had only 40 categories (*ibid.*: 221).

## 4 Discussion and Future Work

In this paper, we describe an on-going project that seeks to develop the first automatic tagger for ScG. We employed supervised methods to develop and evaluate eight different PoS tagging models. Despite the promising results, more work is indicated. Data sparsity is the most likely explanation for the relatively low performance across the models. This is exemplified by the 43% difference between the performance of the simple trigram and unigram models. Considering the size of the our current training set (12.5k tokens) and the granular nature of the tagset (242 discrete categories), it seems unavoidable at present. The majority of tags had less than five instances in the training set, making it difficult for the algorithms to generate useful patterns. We will address this problem soon by including the full corpus, once it has been verified. Subsequently, we will carry out a fine-grained error analysis to determine which PoS features require further development. To improve results, we may integrate a limited amount of morphological analysis, as well a lexical database that has been made available to us (Bauer & Robertson, 2014). Finally, we will be exploring a multi-phase feature disambiguation scheme similar to that described in Acedański (2010: 5).

## Acknowledgements

We would like to thank Prof Mirella Lapata (University of Edinburgh) for reading a draft of this paper and providing helpful comments. Many thanks to project members Dr Sharon Arbuthnot for retagging the corpus and helping to devise the tagset, and Ms Susanna Naismith for her work in verifying and

[8] Since this paper was written, the Brill tagger has achieved 86.8% accuracy (cf 92.5% on word classes only), using an 80k token training sample and 6,460 token test sample.

correcting the corpus. Finally, our appreciation to the Carnegie Trust for the Universities of Scotland and Bòrd na Gàidhlig for their generous financial support.

## References

Szymon Acedański. 2010. A morphosyntactic Brill tagger for inflectional languages. *Advances in Natural Language Processing*. Berlin: Springer Berlin Heidelberg, 3-14

Michael Bauer and William Robertson. 2014. Am Faclair Beag (On-line dictionary). Available at www.faclair.com.

Steven Bird. 2006. NLTK: The natural language toolkit. *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 69-72.

Thorsten Brants. 2000. TnT: a statistical part-of-speech tagger. *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, 224-231

Eric Brill. 1992. A simple rule-based part of speech tagger. *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 112-116

Chungku Chungku, Jurmey Rabgay, and Gertrud Faaß. 2010. *Building NLP resources for Dzongkha: a tagset and a tagged corpus.* Paper presented at the Proceedings of the 8th workshop on Asian language resources, 103-110.

Sandipan Dandapat, Sudeshna Sarkar, and Anupam Basu. 2007. Automatic part-of-speech tagging for Bengali: An approach for morphologically rich languages in a poor resource scenario. *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, 221-224.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. *Multiple Classifier Systems*. Berlin: Springer Berlin Heidelberg, 1-15.

Nancy Dorian. 1973. Grammatical change in a dying dialect. *Language*, 49:413-438.

Zoubin Ghahramani. 2001. An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):9-42.

Zhongqiang Huang, Vladimir Eidelman, and Mary Harper. 2009. Improving a simple bigram HMM part-of-speech tagger by latent annotation and self-training. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Association for Computational Linguistics, 213-216

William Lamb. 2008. *Scottish Gaelic Speech and Writing: Register Variation in an Endangered Language*. Belfast: Cló Ollscoil na Banríona.

Geoffrey Leech. 2005. In Martin Wynne (Ed.), *Developing Linguistic Corpora: a Guide to Good Practice*. Oxford: Oxbow Books, 17-29. Retrieved from http://ahds.ac.uk/linguistic-corpora [accessed 28 April 2014].

Christopher Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing, 12th International Conference, CICLing 2011, Proceedings, Part I.* Lecture Notes in Computer Science 6608. Berlin: Springer Berlin Heidelberg, 171-189

James E. Miller and Regina Weinert. 1998. *Spontaneous Spoken Language: Syntax and Discourse*. Oxford: Clarendon Press.

Csaba Oravecz and Péter Dienes. 2002. Efficient stochastic part-of-speech tagging for Hungarian. *The Proceedings of the Third International Conference on Language Resources and Evaluation (Las Palmas)*, 710-717.

Elmer Ternes. 1982. The grammatical structure of the Celtic languages. In R. Driscoll (Ed.), *The Celtic Consciousness.* Edinburgh: Canongate, 69-78.

Elaine Uí Dhonnchadha. 2009. Part-of-speech tagging and partial parsing for Irish using finite-state transducers and Constraint Grammar. PhD thesis. Dublin City University, School of Computing.

Elaine Uí Dhonnchadha and Joseph van Genabith. 2006. A Part-of-Speech tagger for Irish using finite state morphology and constraint grammar disambiguation. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2241-2244.